OBSERVACIONES DE LA PRACTICA

Estudiante 1 Cod XXXX

Estudiante 2 Cod XXXX

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | 1,4 GHz Intel Core i5 de cuatro núcleos | 1,4 GHz Intel Core i5 de cuatro núcleos |
| Memoria RAM (GB) | 8 GB 2133 MHz LPDDR3 | 8 GB 2133 MHz LPDDR3 |
| Sistema Operativo | macOS Big Sur version 11.2.1 | macOS Big Sur version 11.2.1 |

Tabla 1. Especificaciones de las máquinas para ejecutar las pruebas de rendimiento.

# **Maquina 1**

## **Resultados**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tamaño de la muestra (ARRAYLIST)** | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 1000 | 0.66 | 0.73 | 0.05 | 0,04 | 0,04 |
| 2000 | 2.60 | 2.91 | 0.09 | 0,07 | 0,07 |
| 4000 | 10.71 | 11.75 | 0.20 | 0,15 | 0,14 |
| 8000 | 43.14 | 52.82 | 0.49 | 0,32 | 0,29 |
| 16000 | 184.48 | 241.55 | 1.05 | 0,58 | 0,61 |
| 32000 | Se demora mucho | Se demora mucho | 2.52 | 1,25 | 1,32 |
| 64000 | Se demora mucho | Se demora mucho | 6.90 | 2,82 | 2,78 |
| 128000 | Se demora mucho | Se demora mucho | 14.86 | 6,17 | 5,94 |
| 256000 | Se demora mucho | Se demora mucho | 38.33 | 14,56 | 12,64 |
| 512000 | No alcanzan los datos | No alcanzan los datos | No alcanzan los datos | No alcanzan los datos | No alcanzan los datos |

Tabla 2. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación arreglo.

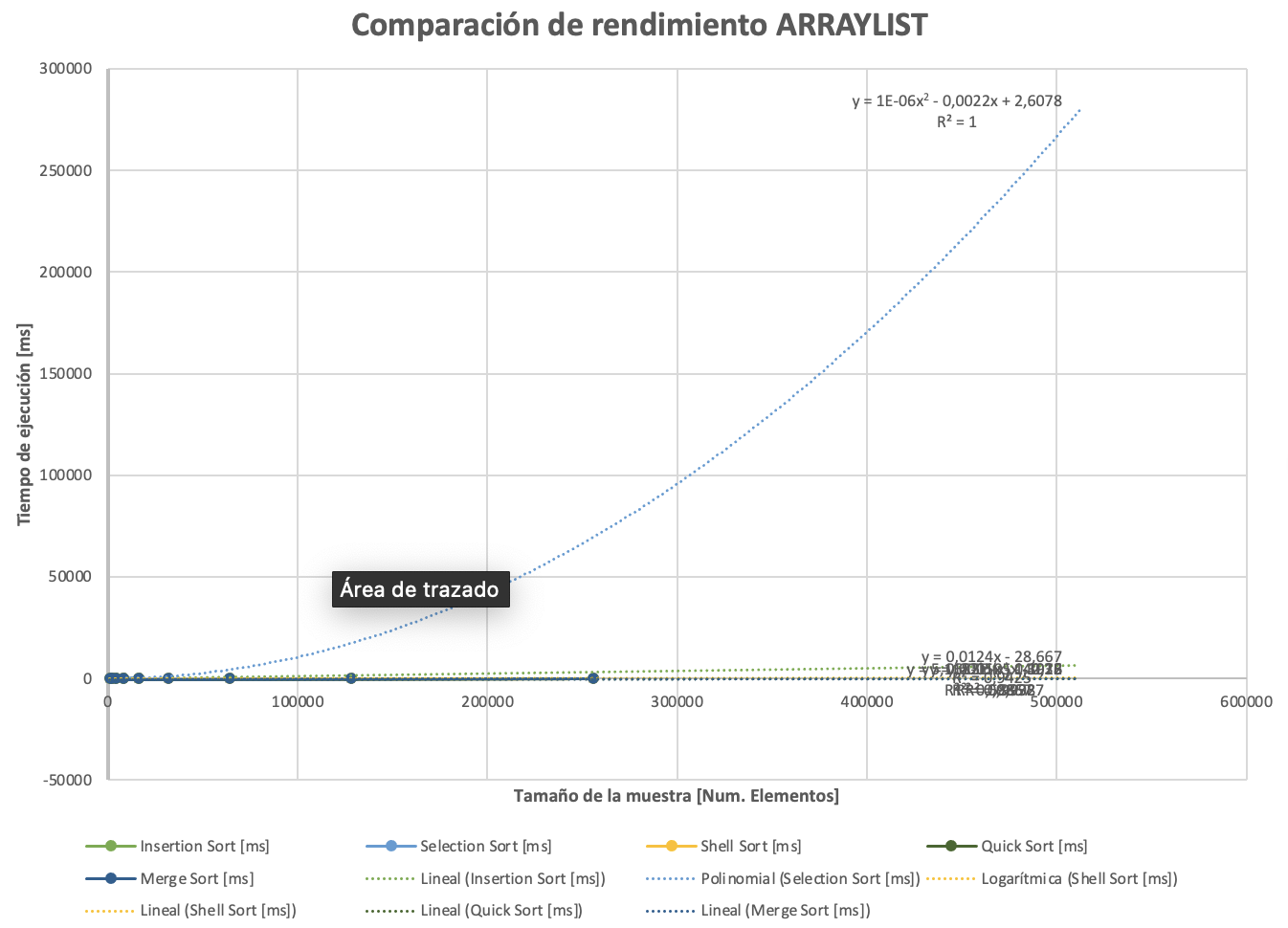
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tamaño de la muestra (LINKED\_LIST)** | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 1000 | 46.36 | 43.33 | 2.39 | 1,63 | 0,27 |
| 2000 | Se demora mucho | Se demora mucho | 11.16 | 8,37 | 1,03 |
| 4000 | Se demora mucho | Se demora mucho | 50.47 | 37,72 | 4.10 |
| 8000 | Se demora mucho | Se demora mucho | Se demora mucho | 171,35 | 16.29 |
| 16000 | Se demora mucho | Se demora mucho | Se demora mucho | Se demora mucho | 67,12 |
| 32000 | Se demora mucho | Se demora mucho | Se demora mucho | Se demora mucho | Se demora mucho |
| 64000 | Se demora mucho | Se demora mucho | Se demora mucho | Se demora mucho | Se demora mucho |
| 128000 | Se demora mucho | Se demora mucho | Se demora mucho | Se demora mucho | Se demora mucho |
| 256000 | Se demora mucho | Se demora mucho | Se demora mucho | Se demora mucho | Se demora mucho |
| 512000 | No alcanzan los datos | No alcanzan los datos | No alcanzan los datos | No alcanzan los datos | No alcanzan los datos |

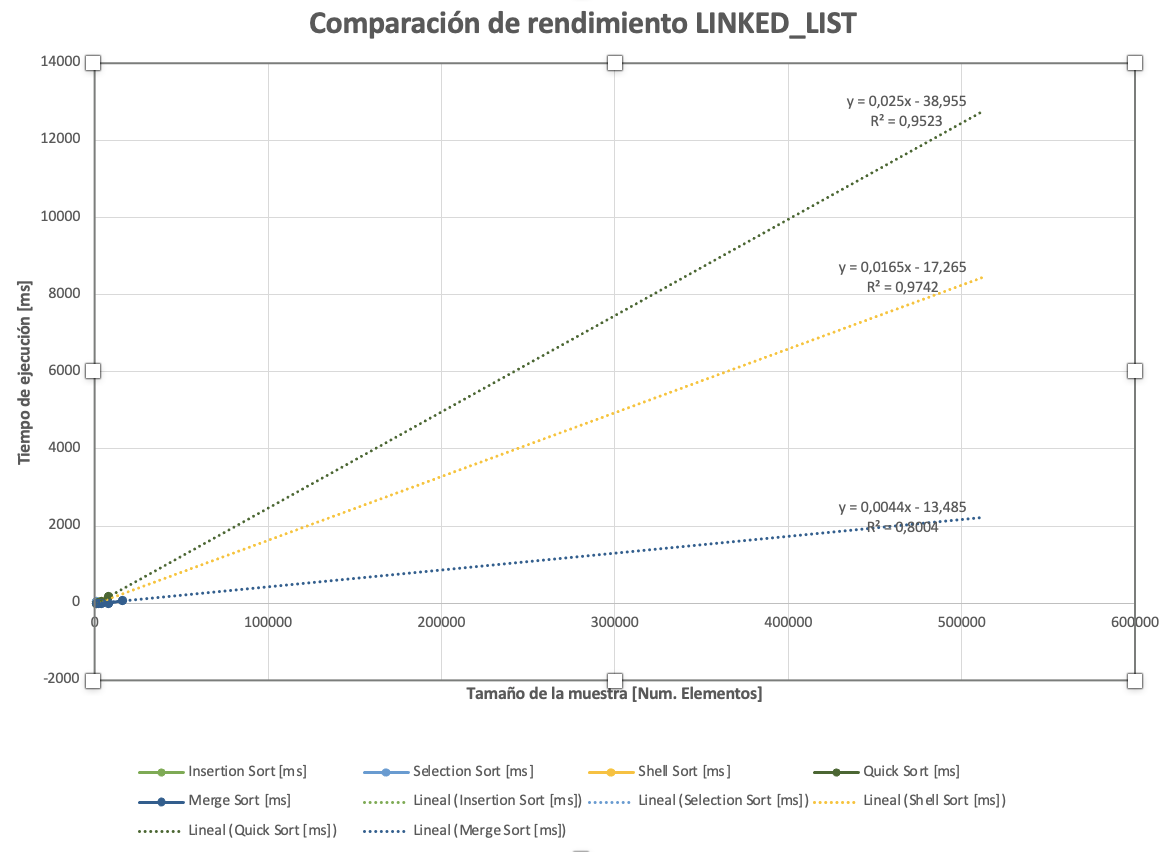
Tabla 3. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Merge sort | 2 | 1 |
| Quick sort | 1 | 2 |

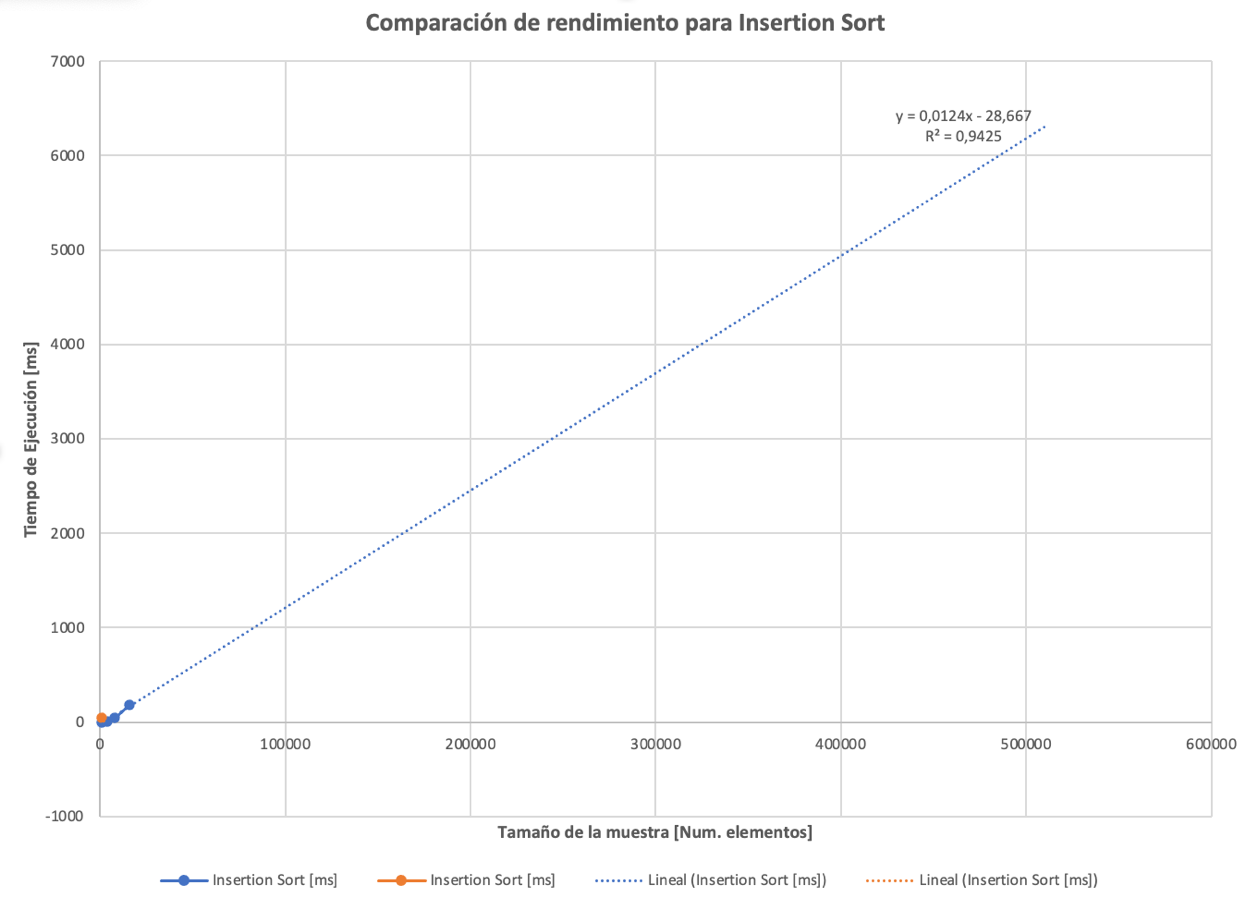
Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

## **Graficas**

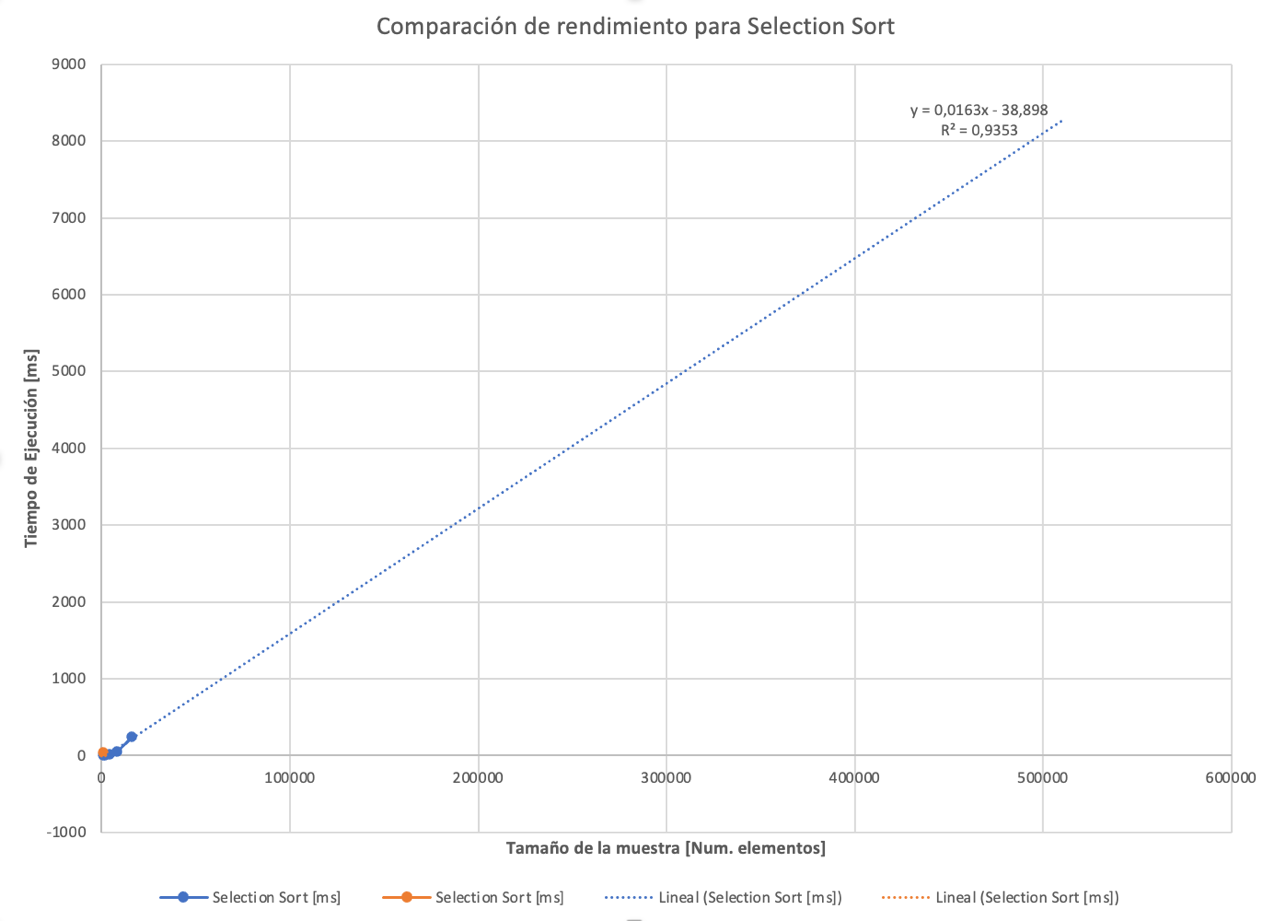
* Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 1.**
  + Comparación de rendimiento ARRAYLIST.
  + Comparación de rendimiento LINKED\_LIST.



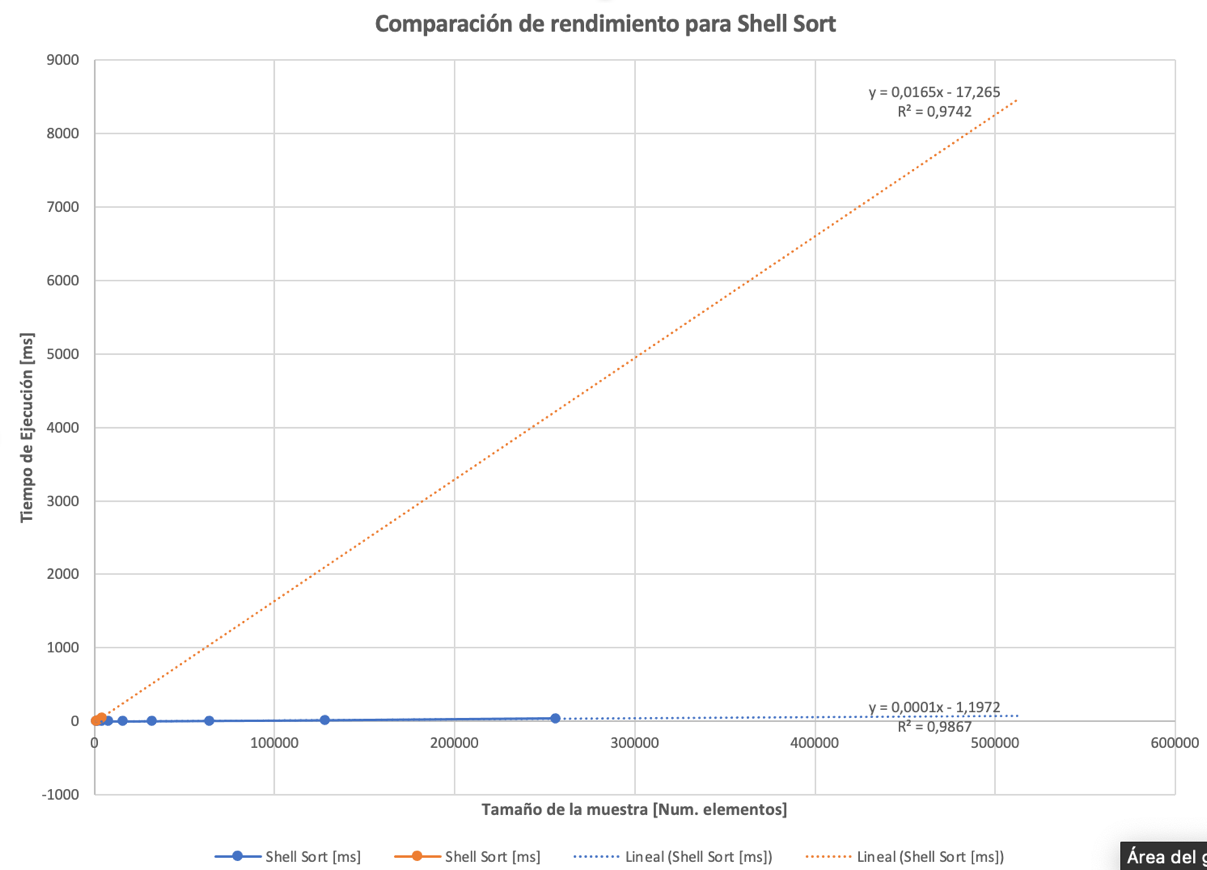
* + Comparación de rendimiento para Insertion Sort.



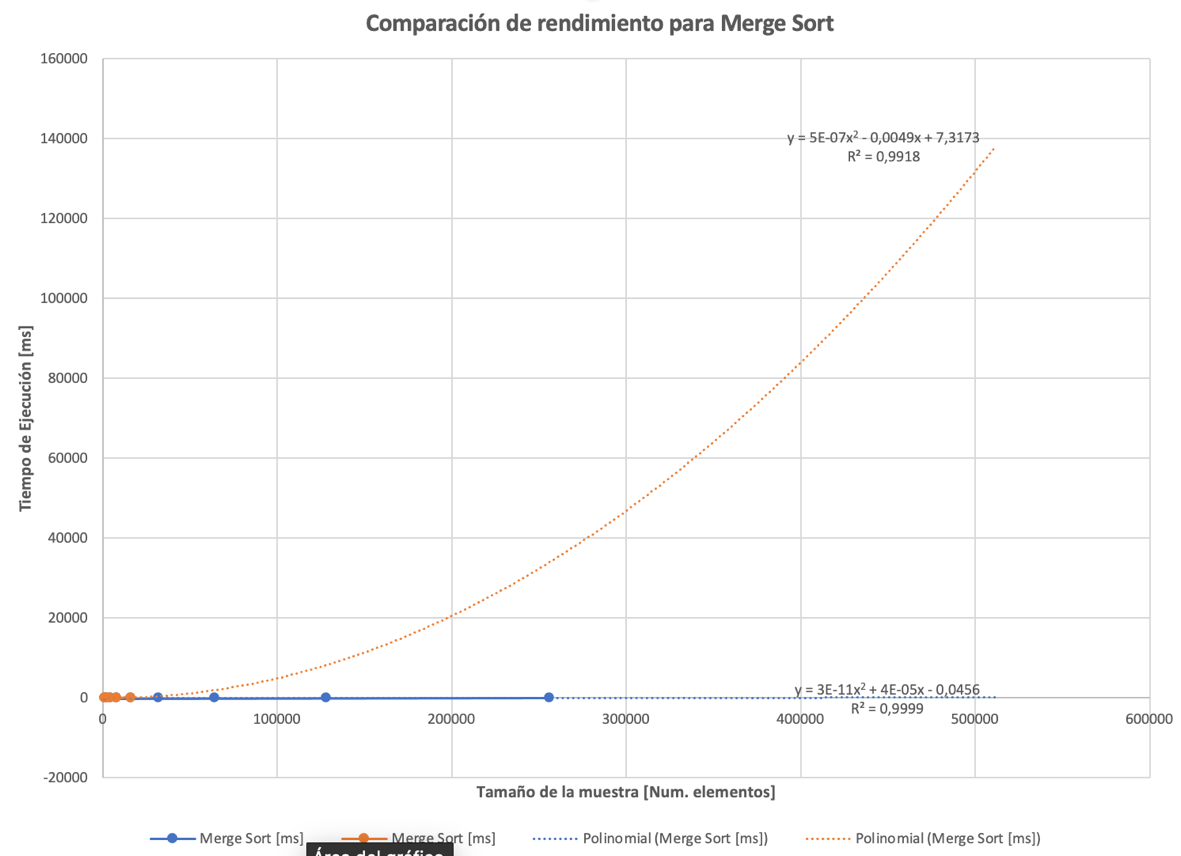
* + Comparación de rendimiento para Selection Sort.



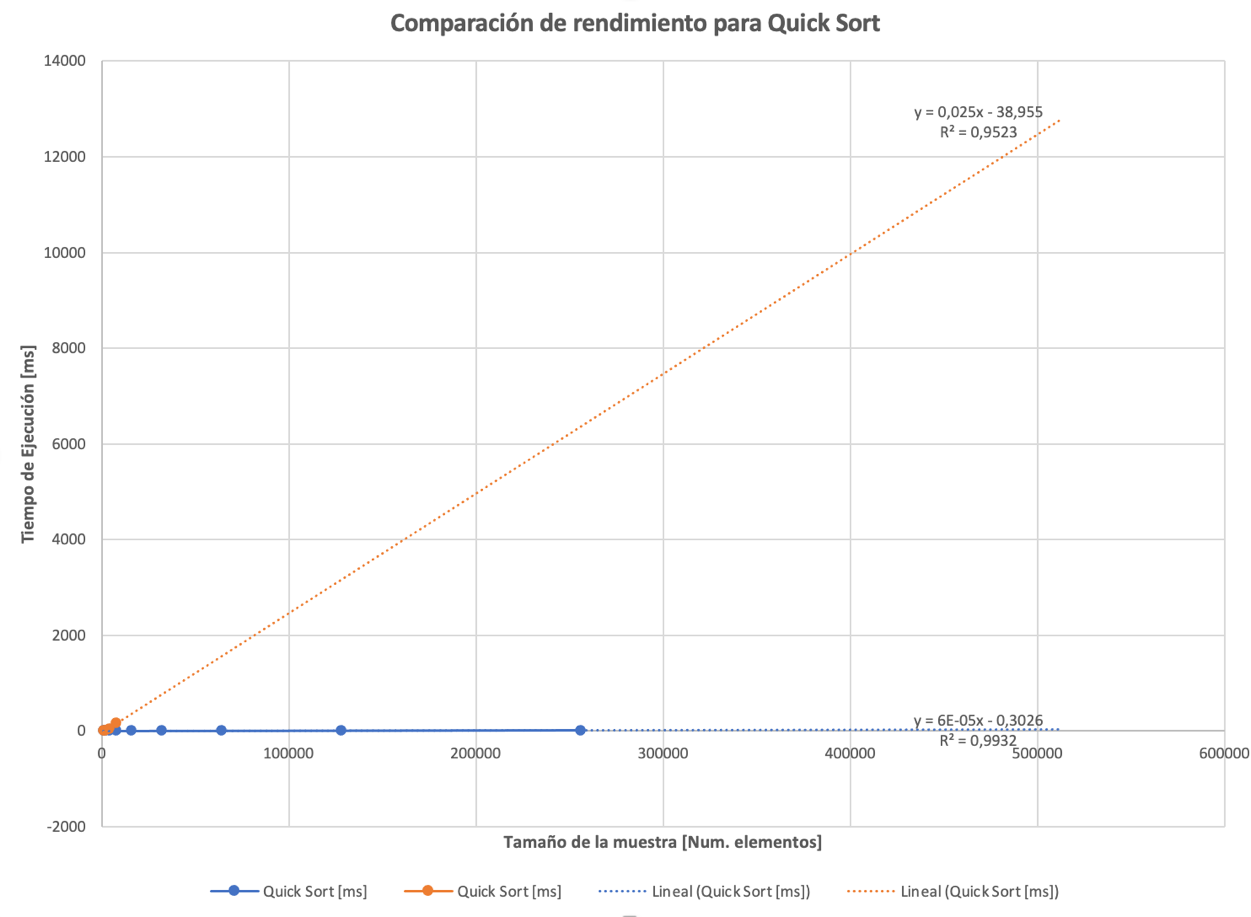
* + Comparación de rendimiento para Shell Sort.



* + Comparación de rendimiento para MergeSort.



* + Comparación de rendimiento para QuickSort.



# **Maquina 2**

## **Resultados**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tamaño de la muestra (ARRAYLIST)** | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 1000 | 0.55 | 0.62 | 0.04 | 0.03 | 0.03 |
| 2000 | 2.26 | 2.51 | 0.09 | 0.06 | 0.07 |
| 4000 | 8.73 | 11.07 | 0.17 | 0.12 | 0.12 |
| 8000 | 40.42 | 48.04 | 0.43 | 0.26 | 0.24 |
| 16000 | 154.26 | 202.01 | 0.89 | 0.47 | 0.52 |
| 32000 | 654.87 | 928.24 | 2.16 | 1.09 | 1.14 |
| 64000 | Se demora mucho | Se demora mucho | 5.25 | 2.46 | 2.48 |
| 128000 | Se demora mucho | Se demora mucho | 11.52 | 5.13 | 5.12 |
| 256000 | Se demora mucho | Se demora mucho | 30.73 | 12.01 | 10.85 |
| 512000 | No alcanzan los datos | No alcanzan los datos | No alcanzan los datos | No alcanzan los datos | No alcanzan los datos |

Tabla 2. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación arreglo.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tamaño de la muestra (LINKED\_LIST)** | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 1000 | 41.14 | 38.32 | 2.21 | 1.54 | 0.25 |
| 2000 | 340.22 | 306.94 | 38.32 | 7.80 | 0.99 |
| 4000 | Se demora mucho | Se demora mucho | Se demora mucho | 35.64 | 3.94 |
| 8000 | Se demora mucho | Se demora mucho | Se demora mucho | 167.48 | 15.20 |
| 16000 | Se demora mucho | Se demora mucho | Se demora mucho | 584.35 | 62.16 |
| 32000 | Se demora mucho | Se demora mucho | Se demora mucho | Se demora mucho | 254.75 |
| 64000 | Se demora mucho | Se demora mucho | Se demora mucho | Se demora mucho | Se demora mucho |
| 128000 | Se demora mucho | Se demora mucho | Se demora mucho | Se demora mucho | Se demora mucho |
| 256000 | Se demora mucho | Se demora mucho | Se demora mucho | Se demora mucho | Se demora mucho |
| 512000 | No alcanzan los datos | No alcanzan los datos | No alcanzan los datos | No alcanzan los datos | No alcanzan los datos |

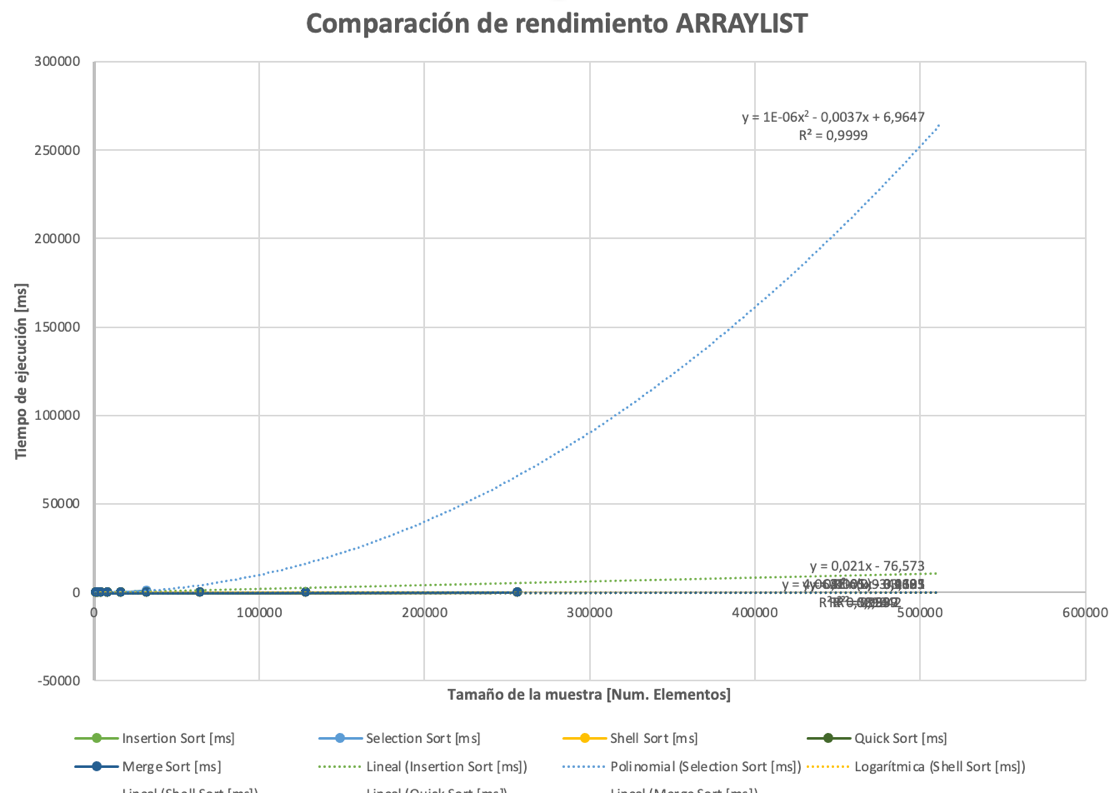
Tabla 3. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Merge sort | 2 | 1 |
| Quick sort | 1 | 2 |

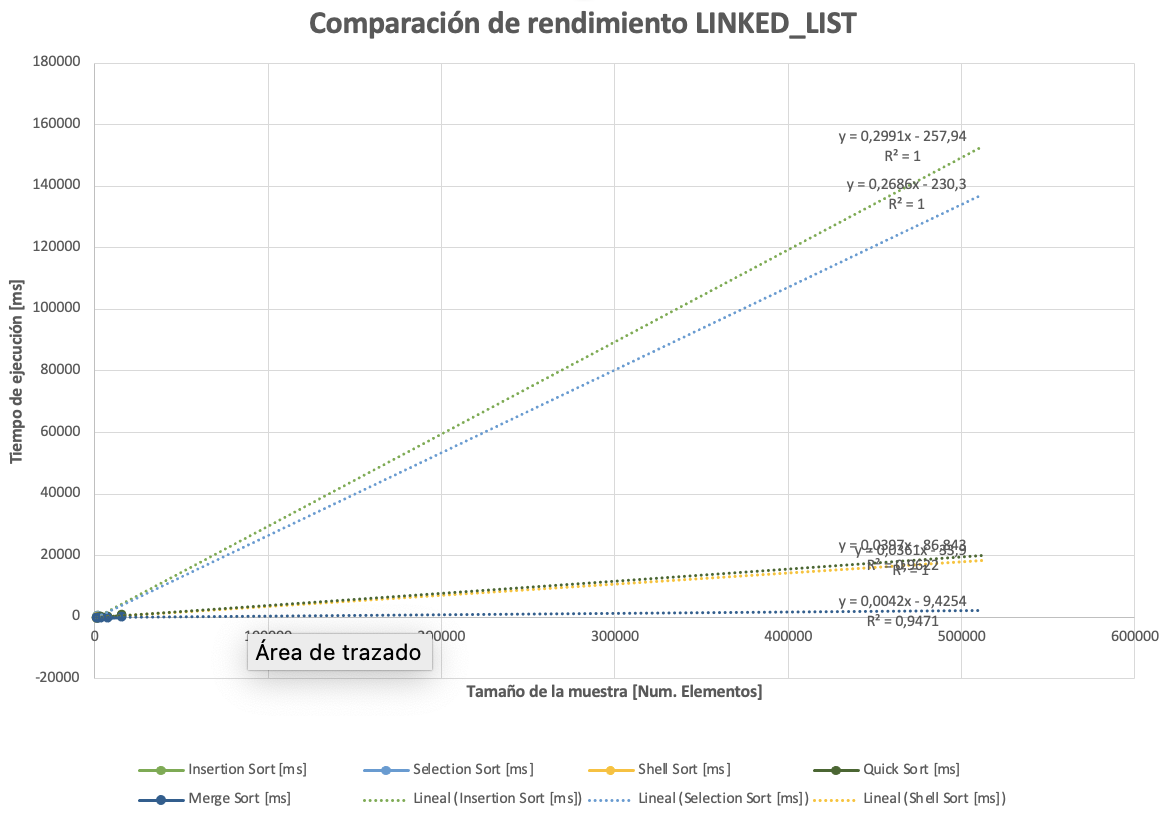
Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

## **Graficas**

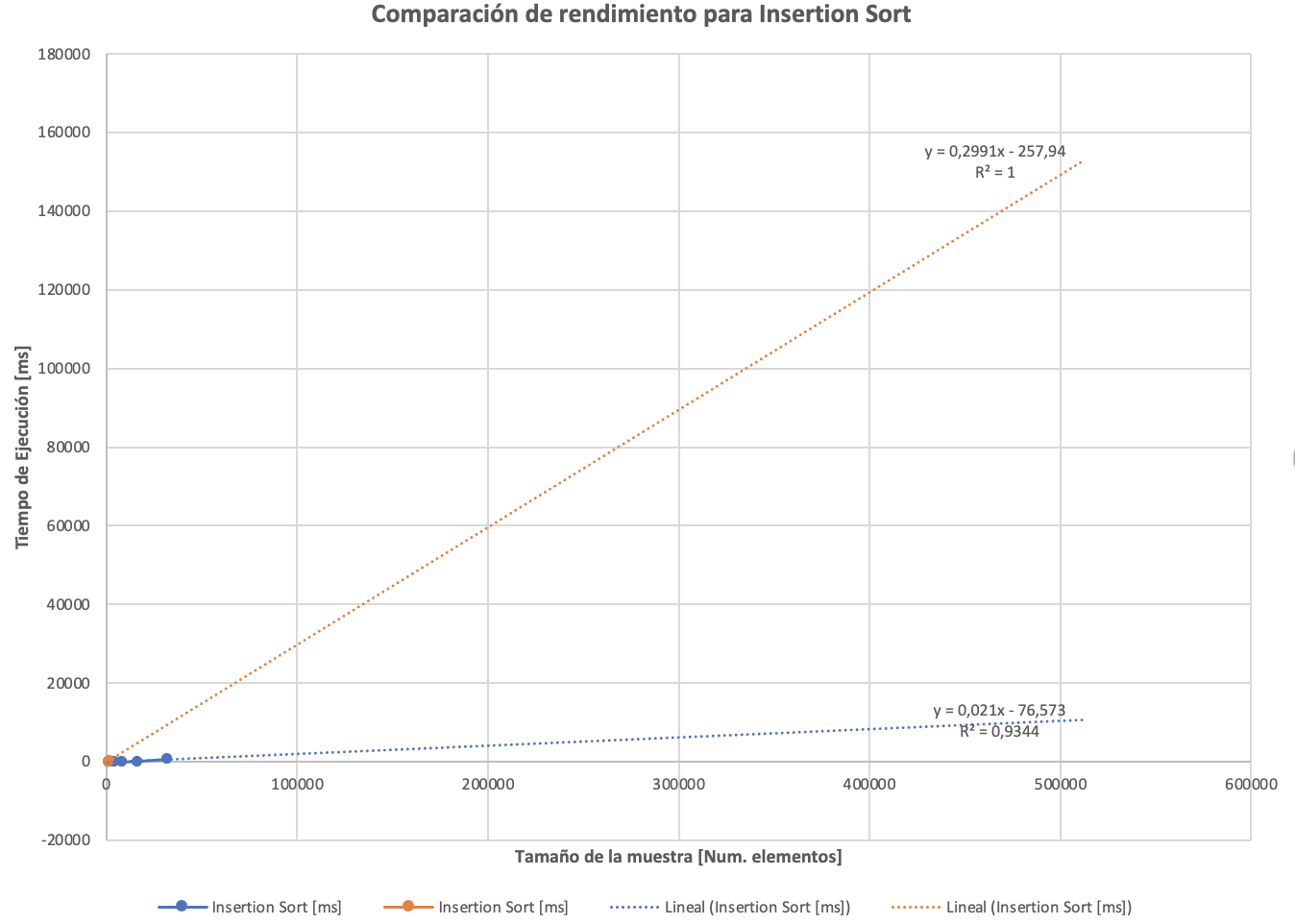
* Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 2.**
  + Comparación de rendimiento ARRAYLIST.



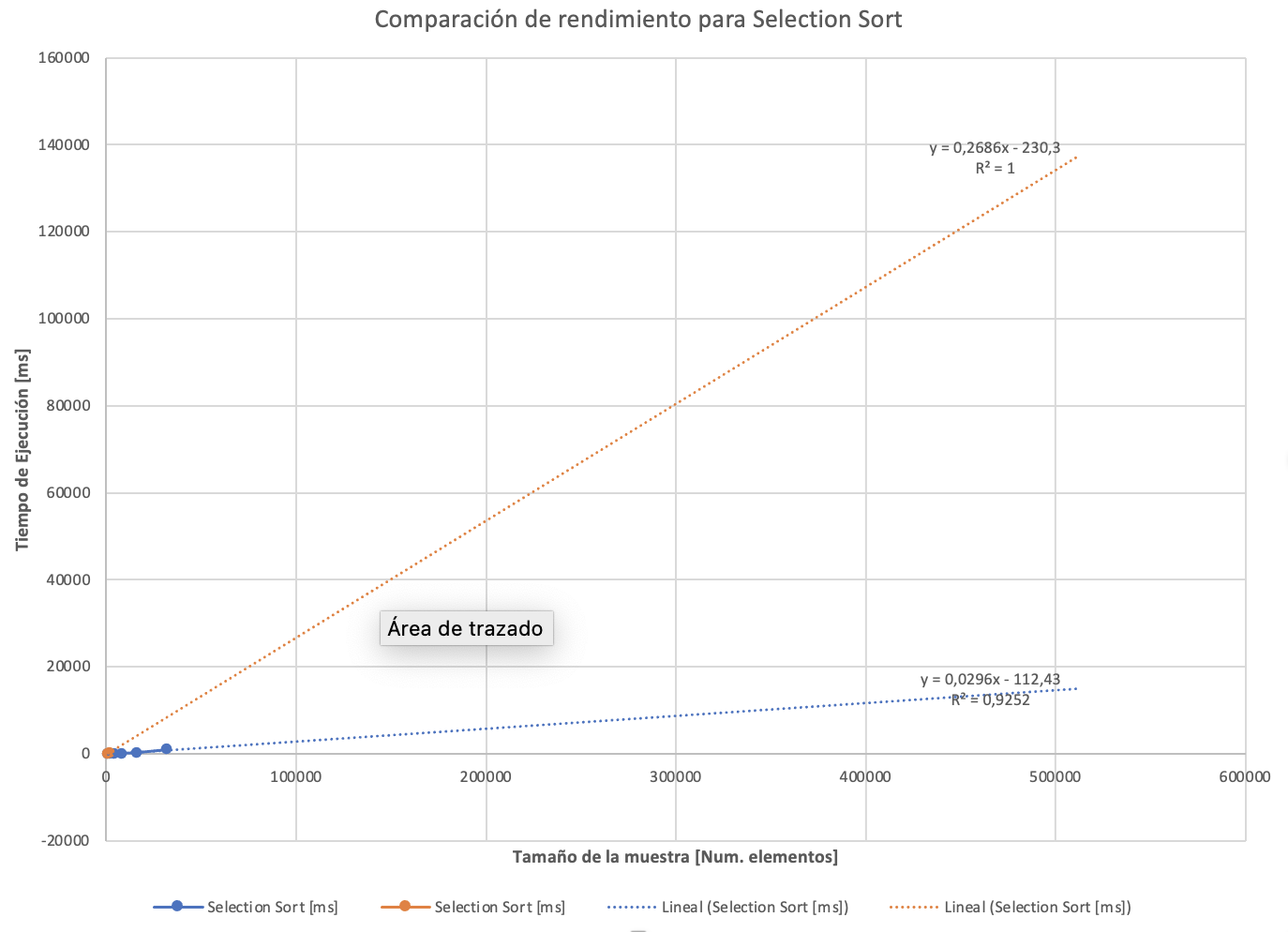
* + Comparación de rendimiento LINKED\_LIST.



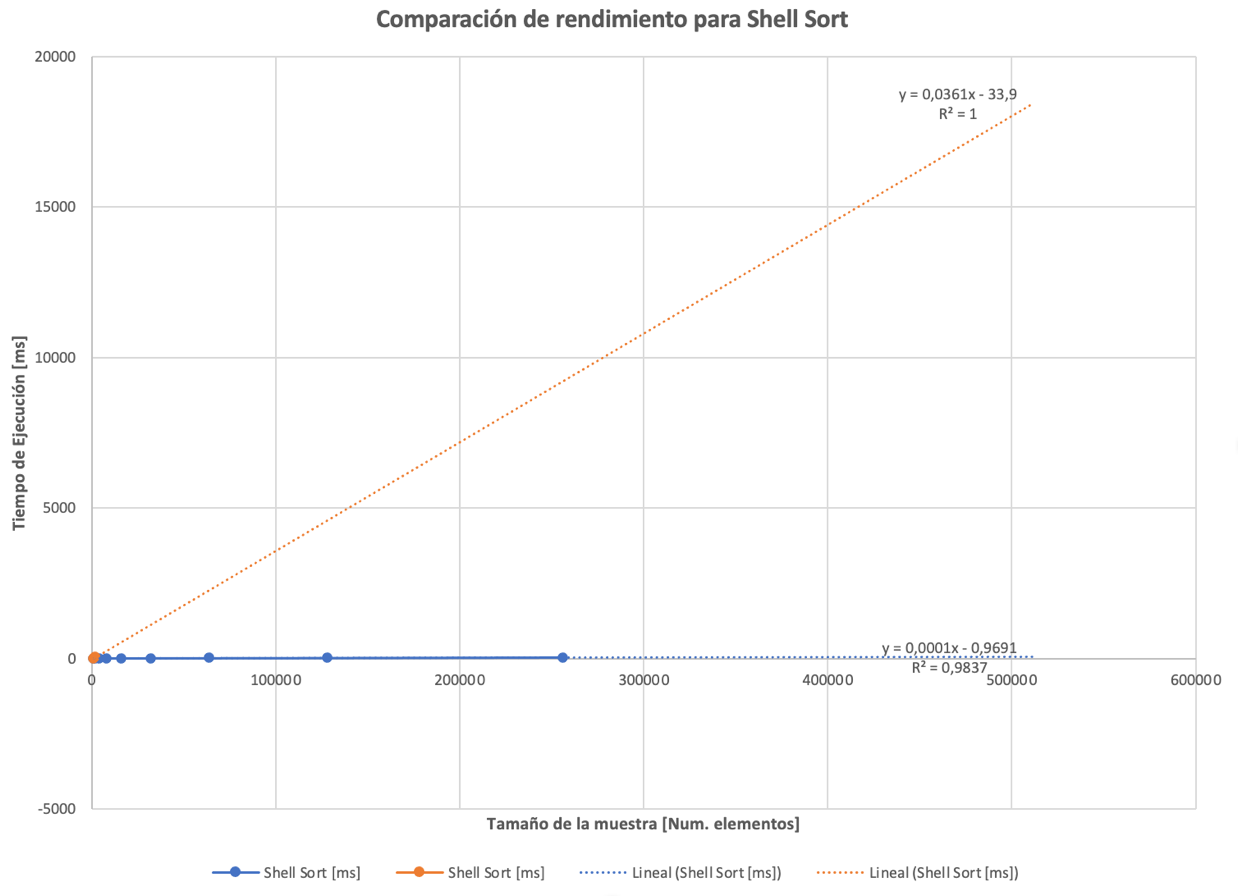
* + Comparación de rendimiento para Insertion Sort.

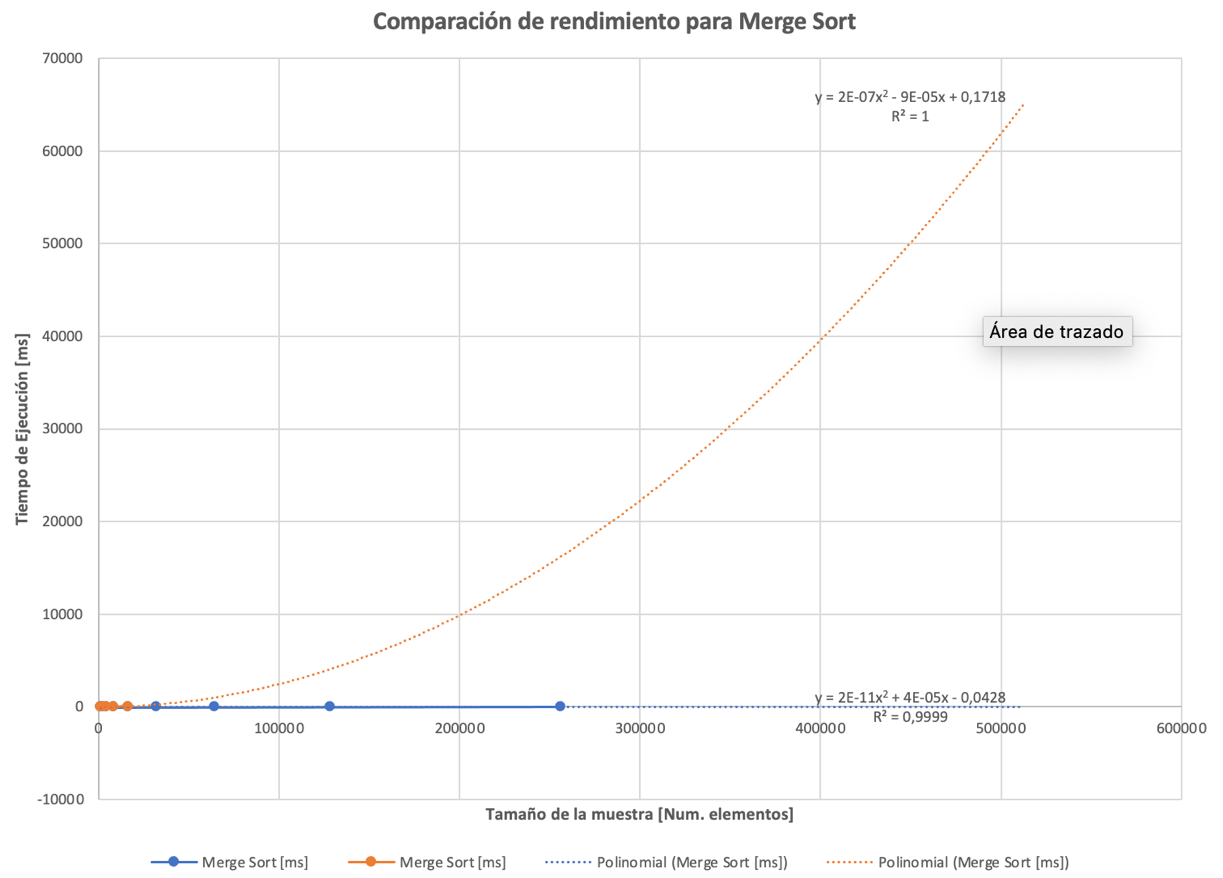


* + Comparación de rendimiento para Selection Sort.



* + Comparación de rendimiento para Shell Sort.



* + Comparación de rendimiento para MergeSort.
  + Comparación de rendimiento para QuickSort.

# 

# **Preguntas de análisis**

1. ¿El comportamiento de los algoritmos es acorde a lo enunciado teóricamente?

* Si, pues segun lo visto en clase el mejor algoritmo es el mergelsort por su orden de complejidad en el peor caso que es nlogn frente a quicksort que es n2.

1. ¿Existe alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas?

* Tenemos el mismo computador así que no existen demasiadas, sin embargo, las hay.

1. De existir diferencias, ¿A qué creen ustedes que se deben dichas diferencias?

* Creeria que a la cantidad de applicaciones abiertas, pues se cuenta con el mismo computador

1. ¿Cuál Estructura de Datos es mejor utilizar si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?

* No hay mejor estructura pues con ARRAY\_LIST es mas rapido el mergesort y con LINKED\_LIST es la rapido el quicksort.

1. Para el caso analizado de ordenamiento de los videos, teniendo en cuenta los resultados de tiempo reportados por todos los algoritmos de ordenamiento estudiados (iterativos y recursivos), proponga un ranking de los algoritmos de ordenamiento (de mayor eficiencia a menor eficiencia en tiempo) para ordenar la mayor cantidad de videos.

* 1.Mergesort 2.Quicksort 3.Shellsort 4.Insertionsort 5.Selctionsort.